接口设计规范说明书

**青岛叁度信息技术有限公司**

**2020年2月**

**版本历史记录**

|  |  |  |  |
| --- | --- | --- | --- |
| **修订号** | **作者** | **日期** | **简要说明** |
| V1.0 | 肖兵 | 2020.2.29 | 建立 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 一、前言

## 范围

根据xxx项目接口要求，完善规范化接口对接内容，结合本次项目的具体建设目标，对以后全部的接口开发交互提供指导建议。

# RESTFul规范

## 协议

接口与前端的通信协议一般使用HTTPS通信协议，确保交互数据的传输安全。

## 域名

全部的接口统一部署在专用域名之下。

例：

<https://api.example.com>

## 接口路径规则

路径又称"终点"（endpoint），表示API的具体网址。  
 在RESTful架构中，每个网址代表一种资源（resource），所以网址中不能有动词，只能有名词，而且所用的名词往往与数据库的表格名对应。一般来说，数据库中的表都是同种记录的"集合"（collection），所以API中的名词也应该使用复数。

例：

<https://api.example.com/oms/orders> //查询全部订单

## HTTP请求方式

对于资源的具体操作类型，由HTTP动词表示。  
 常用的HTTP动词有下面四个（括号里是对应的SQL命令）。  
 GET（SELECT）：从服务器取出资源（一项或多项）。  
 POST（CREATE）：在服务器新建一个资源。  
 PUT（UPDATE）：在服务器更新资源（客户端提供改变后的完整资源）。  
 DELETE（DELETE）：从服务器删除资源。

例：

GET /order：列出所有订单  
 POST /order：新建一个订单  
 GET /order/ID：获取某个指定订单的信息  
 PUT /order/ID：更新某个指定订单的信息  
 DELETE /order/ID：删除某个订单

## 过滤信息

如果记录数量很多，服务器不可能都将它们返回给用户。API应该提供参数，过滤返回结果。

例：

下面是一些常见的参数。  
 ?limit=10：指定返回记录的数量  
 ?offset=10：指定返回记录的开始位置。  
 ?page=2&per\_page=100：指定第几页，以及每页的记录数。  
 ?sortby=name&order=asc：指定返回结果按照哪个属性排序，以及排序顺序。  
 ?producy\_type=1：指定筛选条件

注：以上参数使用拼接URL参数的方式进行传参，应注意避免敏感数据的出现

## 传入参数

传入参数分为3种类型：

地址栏参数  
 \* restful 地址栏参数 /oms/order/122 122为订单编号，获取订单为122的信息  
 \* get方式的查询字串 见过滤信息小节  
 请求body数据  
 request header 一般都是用于OAuth认证的途径

## 返回数据

为了保障前后端的数据交互的顺畅，必须使用规范数据的返回，并采用固定的数据格式封装。

例：

{  
 code:0, //接口的执行状态，=0表示正常，!=0表示失败

msg:’’, //错误描述信息，要求该信息尽量做到调用方可以直接显示给用户  
 data:{} //接口的返回数据  
}

# 三、完整接口请求流程示范

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 接口示例：以下示例存在参数与实际业务不符情况，请读者忽略   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 服务名 | | 获取某一用户的订单列表 | | | | 说明 | | 分页查询，默认每页查询20条 | | | | 实际URL | | http://api.example.com/oms/orders?pageIndex=0&pageSize=20 | | | | Header | | token=\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* | | | | 接口类型 | | POST | | | | GET参数 | | | | | | 字段名 | 类型 | | 是否必须 | 说明 | | pageIndex | Int | | 是 | 要查询第几页，从0开始，该字段由调用  方维护 | | pageSize | Int | | 否 | 设置每页查询多少条数据，默认20 | | POST参数： | | | | | | 字段名 | 类型 | | 是否必须 | 说明 | | beginTime | String | | 否 | 查询条件，订单的创建时间 | | endTime | String | | 否 | 查询条件，订单的创建时间 | | Status | Int | | 否 | 订单状态，默认查询全部状态，  1:正常状态  2:完成状态 | | province | Int | | 是 | 省份代码 | | 返回值： | | | | | | 字段名 | 类型 | | 是否必须 | 说明 | | code | Int | | 是 | 请求状态 | | msg | String | | 是 | 错误信息 | | data | Object | | 否 | 返回数据 | | 输入数据格式举例：  {  "beginTime": "2019-02-03",//查询条件，查询创建时间在该时间之后的订单  "endTime": "2019-06-06", //查询条件，查询创建时间在该时间之前的订单  "status": 1, //订单状态  "province": 209 //省份代码  } | | | | | | 返回数据格式举例：  {  code:0, //接口的执行状态，=0表示正常，!=0表示失败  msg:’’, //错误描述信息，要求该信息尽量做到调用方可以直接显示给用户  data:{ //接口的返回数据    }  } | | | | | |

  注：虽然网络传输时均是以字节的形式进行传输，但不同的编码格式生成的字节是不同的，因而需要对此进行统一。如果双方系统的编码格式不同则在进行数据处理时必须进行转码。